加微信：642945106 发送“ 赠送 ”领取赠送精品课程
发数字“ 2 ”获取众筹列表

01 | SRE迷思：无所不能的角色？还是运维的升级？

2020-03-18 赵成

SRE实战手册

进入课程 >

讲述：赵成
时长 11:00　大小 10.09M

你好，我是赵成。

作为这个课程的第一讲，我先从实践的角度，和你聊聊应该怎么理解 SRE。

为什么要强调是实践的角度呢？

开篇词里我们就提到过，有人认为 SRE 就是一个岗位，而且是一个具备全栈能力的岗位，只要有这么一个人，他就能解决所有稳定性问题。这还只是一种理解，而且这个理解站在管理者的角度。

也有人站在运维人员的角度，认为做好 SRE 主要是做好监控，做到快速发现问题、快速找到问题根因；还有人站在平台的角度，认为做好 SRE 要加强容量规划，学习 Google 做到完全自动化的弹性伸缩；甚至还有人认为，SRE 就是传统运维的升级版，把运维自动化做好就行了。

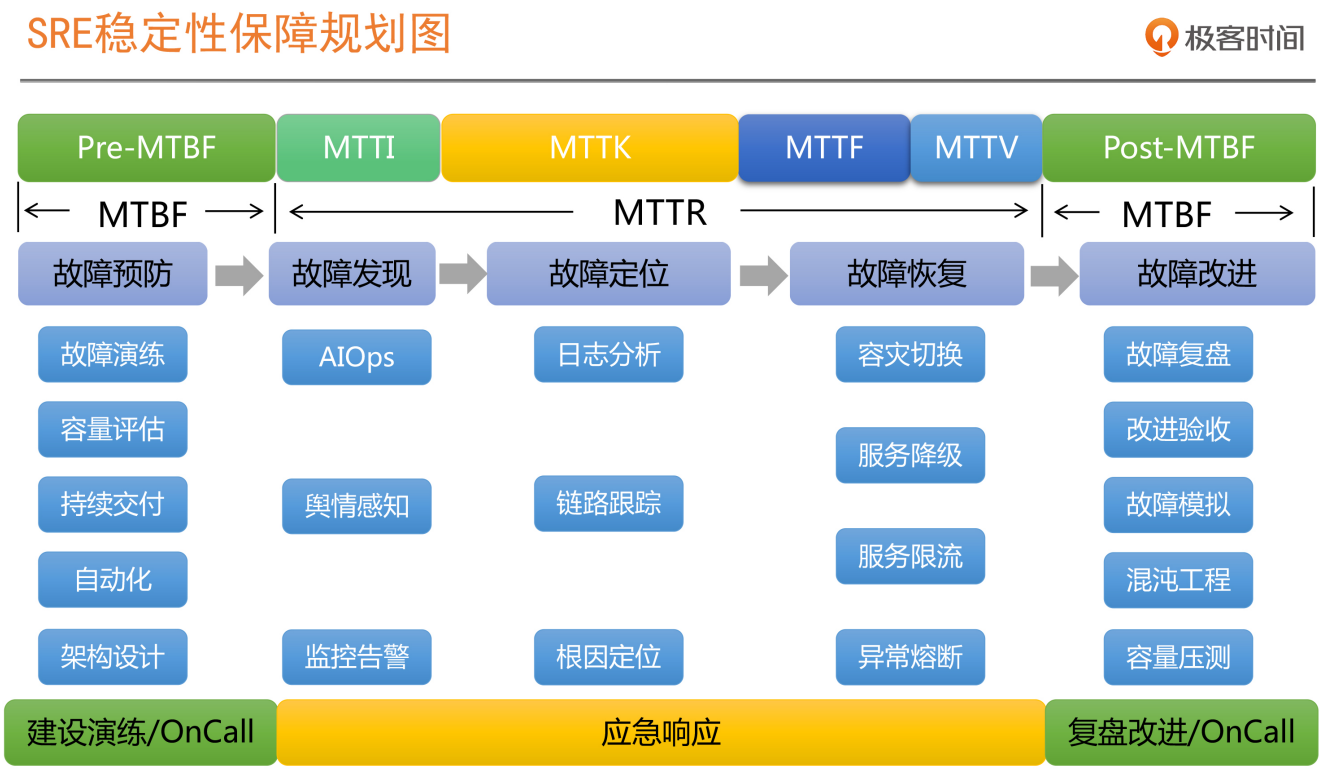
你看，其实不同的人站在不同的角度，对 SRE 的理解就会天差地别，但是好像又都有各自的道理。

所以，我特别强调实践的角度，我们不站队，就看真实的落地情况。我总结了一下从实践角度看 SRE 的关键点，就一个词：**体系化。SRE 是一套体系化的方法，我们也只有用全局视角才能更透彻地理解它。**

好了，下面我们就一起来看怎么理解 SRE 这个体系化工程。

SRE，我们应该怎么来理解它？

我先给你分享一张图，这是结合我自己团队的日常工作，做出来的 SRE 稳定性保障规划图。



我们最初画这张图是为了提高故障处理效率，将每个阶段可以做的事情填了进去，并在实践中不断补充完善，最终形成了我们探索 SRE 的框架图。你应该也发现了，这里面很多事情都很常见，比如容量评估、故障演练、服务降级、服务限流、异常熔断、监控告警等等。

这就是为什么我一直给你打气，说 SRE 并不神秘。学习 SRE，我们可以有非常熟悉的抓手。但是，我们不能停留在向 Google 或其他大厂学习具体的技术经验上，而是更应该学习如何将这些技术有机地结合起来，形成一套稳定性体系，**让体系发挥出力量**，告别只发挥某项技术的能力。

同时，结合这些具体的事情，你应该明白，这些工作并不是某个人、某个角色，甚至是某个团队就可以单枪匹马完成的。比如这里的很多事情要依赖运维自动化，像容量扩缩容，必然会与运维团队有交集；如果做得再弹性一些，还需要与监控结合，就要与监控团队有合作；同时还可能依赖 DevOps 提供持续交付、配置变更，以及灰度发布这些基础能力，就要与开发和效能团队有交集。

这些能力之间的相互依赖，就决定了**从职能分工上，SRE 体系的建设绝不是单个岗位或单个部门就能独立完成的，必然要求有高效的跨团队组织协作才可以。**

防止断更 请备必加
首发微信：17161143665

所以，不要想着设定一个 SRE 岗位，就能把稳定性的事情全部解决掉，这明显不现实。你应该从体系的角度出发，设置不同的职能岗位，同时还要有让不同角色有效协作的机制。

从另一个角度来讲，如果当前我们的稳定性建设还仅仅聚焦在某个或某些具体的技术点上，每个角色和团队之间的工作相对还是独立、各自为战的，那就说明 SRE 体系和组织的真正威力还没有发挥出来。

所以，如果你已经是这些领域的实践者，比如你是一个运维工程师，或者是一位监控开发工程师，又或者是一个 DevOps 开发工程师，我建议你除了负责当前的事情外，应该更多地关注一下 SRE 全局视图。这样会帮助你深入了解 SRE 岗位所需要的技术能力，进而提升你的平台架构能力。

要知道，如果严格遵循 Google 的要求，SRE 岗位对技术要求是非常高的。只有具备了全面的技术能力，拥有了全局架构的思维，你才会在 SRE 的道路上走得更远。

总结一下，从实践角度来看，SRE 的力量不能通过一个岗位、一个或几个技术就发挥出来。SRE 是一个体系化工程，它需要协同多个部门、多项技术。

做好 SRE 的根本目的是什么?

认识到 SRE 是个体系化的事儿还不够，我们还可以“以终为始”，看看 SRE 最后要达成什么样的目标，以此来加深对它的理解。

SRE 做了这么多的事情，最后的目的是什么呢?

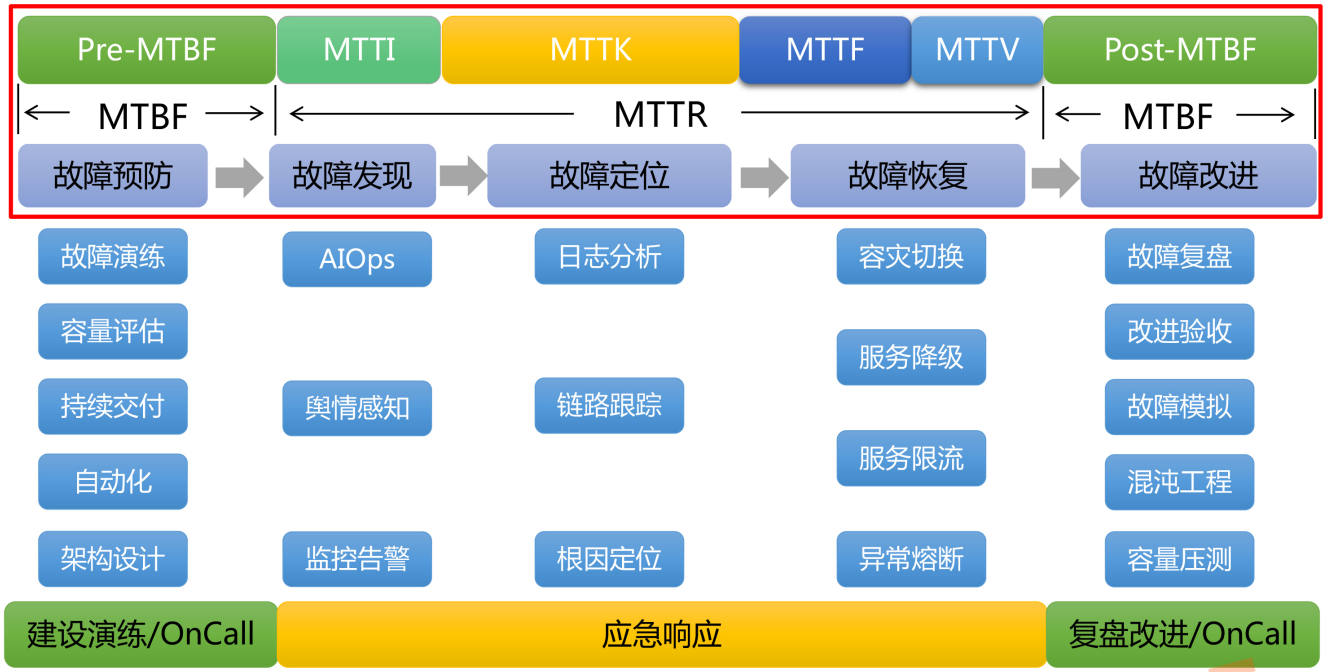
这个答案很明显嘛，就是提升稳定性。但是怎样才算提升了稳定性呢？要回答这个问题，我们有必要来讨论下稳定性的衡量标准。

从业界稳定性通用的衡量标准看，有两个非常关键的指标：

MTBF，Mean Time Between Failure，平均故障时间间隔。

MTTR，Mean Time To Repair， 故障平均修复时间。

还来看前面的 SRE 稳定性保障规划图，你会发现我们把整个软件运行周期按照这两个指标分成了两段。通俗地说，MTBF 指示了系统正常运行的阶段，而 MTTR 则意味着系统故障状态的阶段。



到了这里，我们也就明白了，如果想提升稳定性，就会有两个方向：**提升 MTBF**，也就是减少故障发生次数，提升故障发生间隔时长；**降低 MTTR**，故障不可避免，那就提升故障处理效率，减少故障影响时长。

拼课微信：17161143665

你想，如果我们把故障发生时间的间隔变长，并将故障影响的时间减少，系统稳定性是不是自然就提升了呢？答案是显然的。

从 SRE 稳定性保障规划图中，可以看出 MTTR 可以细分为 4 个指标：MTTI、MTTK、MTTF 和 MTTV。我把它们的具体解释做了一张图，你可以保存下来，有时间就琢磨一下。这 4 个指标，你要烂熟于心。

MTTR细分为4个指标

极客时间

- **MTTI（Mean Time To Identify，平均故障发现时间）**，也就是从故障实际发生，到我们真正开始响应的时间。这个过程可能是用户或客服反馈、舆情监控或者是监控告警等渠道触发的。
- **MTTK（Mean Time To Know，平均故障认知时间）**，更通俗一点，可以理解为我们常说的平均故障定位时间。这个定位指的是root cause，也就是根因被定位出来为止。
- **MTTF（Mean Time To Fix，平均故障解决时间）**，也就是从知道了根因在哪里，到我们采取措施恢复业务为止。这里采取的手段就很多了，比如常见的限流、降级、熔断，甚至是重启。
- **MTTV（Mean Time To Verify，平均故障修复验证时间）**，就是故障解决后，我们通过用户反馈、监控指标观察等手段，来确认业务是否真正恢复所用的时间。

现在，我们再来看 SRE 稳定性保障规划这张图，你就会理解，为什么要把所做的事情分组分块呈现。目的就是区分清楚，**我们做的任何一件事情、开发的任何一套系统、引入的任何一个理念和方法论，有且只有一个目标，那就是“提升 MTBF，降低 MTTR”**，也就是把故障发生时间的间隔变长，将故障影响的时间减少。

比如，在 Pre-MTBF 阶段（无故障阶段），我们要做好架构设计，提供限流、降级、熔断这些 Design-for-Failure 的服务治理手段，以具备故障快速隔离的条件；还可以考虑引入混沌工程这样的故障模拟机制，在线上模拟故障，提前发现问题。

在 Post-MTBF 阶段，也就是上一故障刚结束，开启新的 MTBF 阶段，我们应该要做故障复盘，总结经验，找到不足，落地改进措施等。

在 MTTI 阶段，我们就需要依赖监控系统帮我们及时发现问题，对于复杂度较高和体量非常大的系统，要依赖 AIOps 的能力，提升告警准确率，做出精准的响应。

同时 AIOps 能力在大规模分布式系统中，在 MTTK 阶段也非常关键，因为我们在这个阶段需要确认根因，至少是根因的范围。

你看，我们做了很多事情， 新的理念和方法出来后也特别愿意尝试，就是因为有明确的目标，所有工作的开展都是围绕这个核心目标而去的。

好了，按照以终为始的思路，SRE 要实现的目标就是“提升 MTBF、降低 MTTR”，从这个角度出发，我们再次认识到一定要**把 SRE 作为一个体系化的工程来建设**，因为单纯在某些技术点上发力是没有多大意义的。

总结

今天我要分享的内容就到这里了。总结一下，你需要记住下面这两点。

第一，我们需要从全局的角度去理解 SRE。SRE 一定是靠整个技术和组织体系发挥作用的，单纯从某个技术点或环节出发，都无法呈现出效果，也就是说 **SRE 一定要从全局考虑，体系一定要“配套”。**

第二，SRE 的目的，本质上是减少故障时间，增加系统正常运行时间，也就是 **“减少故障，提升 MTBF；同时，提升故障处理效率，降低 MTTR”**。SRE 要做的所有事儿，都是为这两个目标服务的。

我想无论你是团队负责人、架构师，还是一线的技术专家，有了这样的全局视角后，就会知道接下来应该从何入手，以及如何与其他团队协作来构建这样的体系。

思考题

开篇词中我们提到过大家对 SRE 的困惑，比如 SRE 和 DevOps 都是很好的方法论，我应该怎么选？到底哪个更适合我？今天我们讲了 SRE 应该怎么理解，我想对这个问题你应该有答案了。那就请你来说一说，SRE 和 DevOps 到底哪个更适合你？它们有什么区别和相同点？

请你在留言区说出自己的思考，也欢迎你把今天的内容分享给身边的朋友，和他一起讨论。

我们下节课见。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　开篇词 | SRE是解决系统稳定性问题的灵丹妙药吗?

下一篇　02 | 系统可用性：没有故障，系统就一定是稳定的吗?

精选留言 (22)

写留言

无聊的上帝
2020-03-18

说下我的感受。
DevOps主要是以驱动价值交付为主，搭建企业内部的功效平台。
SRE主要需要协调多团队合作来提高稳定性。
例如：
- 与开发和业务团队落实降级...

展开

作者回复: 分析地恰到好处，你找到了看待这个问题的正确角度。

12

云峰

2020-03-20

赵老师，好。SRE是否从项目开始就需要参与系统架构设计？如果只是在项目上线运行后才接触，遇到架构不合理的地方如何处理？

展开

作者回复: 肯定是越早参与越好，并不一定参与设计本身，但是要知道再哪里提出稳定性要求。比如一个促销场景，要知道可能流量是怎么样的，限流措施要设定在哪些接口上，限流量多大，通过什么方式验证等等，通过这种提要求的方式，倒逼业务开发和架构师思考设计和编码。

1

乐迪_张建

2020-03-20

个人觉得DEVOPS目的是更快的创新和更好的客户体验。SER的目的是最快的速度恢复故障！不管是AWS的DEVOPS还是GG的SRE，适合的才是最好的！都是在慢慢通往这个道路上。最后隐隐约约:SRE和DevOps它有，它无:它无，它有！不知道用语言怎么表达

展开

作者回复: 感觉很准奥，可以看看其他评论内容，这个问题基本就有答案了。

1

子杨

2020-03-19

听成哥这么一说，顿时有种豁然开朗的感觉。降低 MTTR，提升 MTBF。而 MTTR 里面，MTTI 一定要快。不过这里有一点不太理解，平常都是要以业务恢复为第一优先级的，这时候可能回滚变更操作就非常重要了，然后再去定位根因，先定位根因再去恢复业务，这个是有适用场景的吧。

展开

作者回复: 非常棒的问题，而且“业务恢复”永远都是第一优先级，你的理解非常正确。

这里提到是正常流程，但是实践中我们一定会根据实际情况来活学活用，所以，是不是要定位出根因，还是定位出根因范围就要看取舍了，有时我们大致定位出根因范围，然后就要开始执行响应的恢复和隔离措施了。

有点类似当前我们正在经历的疫情，虽然我们知道是新型病毒，但是根源来自哪里并不清楚，而且也没有预防和有效的治愈良药，这时最有效的办法就是提前发现，然后隔离，这种就不可能等着根因定位清楚，再采取措施。

1

海格
2020-03-19

SRE解决运维领域的故障目标，DevOps更偏向于为价值导向的效率目标，但是这个又是你中有我，我中有你，互相成就的一个过程，在实践SRE体系过程中，不可避免的要使用到一些DevOps中的一些技术，方法论，组织文化等，通过这些，达成一致目标。~~~

展开

作者回复: 理解地非常正确!

1

萨神
2020-03-18

class SRE implements DevOps
可以简单理解为DevOps 是一种接口，但是没说怎么实现，SRE 提供了一种视角，这么做在Google 成功过，可以结合自己企业的特点去实现DevOps 这个接口，做有自己特色的 ‘SRE’ 即可。

展开

作者回复: SRE是DevOps的一项最佳实践。

1 1

海峰
2020-03-18

看赵老师之前的书讲：SRE是 用软件工程的方法重新设计和定义运维。

作者回复: SRE的一个理念，非常关键，一定要通过技术手段解决运维问题，而不是人肉投入。

1

糖糖
2020-03-18

SRE 要求对公司业务架构要有一个宏观的了解

展开

作者回复: 你讲到一个非常重要的点，SRE要想做好，必须要对公司业务有全局的了解，甚至是非常深入的了解

2 1

西湖游侠
2020-03-22

说的很清楚，SRE就是一个体系化工程

展开

作者回复: 谢谢，多交流^_^

Daniel
2020-03-21

赵老师，其实从实际问题处理的情况看，根因分析放在故障定位是不太合适的，这样会耽误故障恢复的，重点应该放在如何恢复。比如很多时候我们找到异常组件进行隔离就可以了，然后在事后分析那个组件的异常原因。

展开

作者回复: 非常好的问题啊，而且你的分析是正确的。

关于这个问题，我在另外一位同学的留言中答复了，你可以先看一下，在06篇中，我也会讲到这个问题，到时可以看一下。

1

刘冲
2020-03-21

我想问个问题，怎么算一个故障？客户投诉或者自己发现？或者有些问题可能是故障有些可能不是故障，粒度怎么界定

展开

作者回复: 这是个很好的问题，说明已经思考的很深入了。

关于这部分内容，提前预告下，在SRE中我们会用Error Budget来细化度量，我们将在接下来的04篇文章中会介绍。

于加硕
2020-03-21

赵老师你好，听完本篇我又重新听了开篇。您接下来所讲的内容，应该都是基于团队已经在SRE实践的道路上。1.那么我们该如何判断某条业务线是否值的推行SRE体系呢？(业务的背景大致可以理解为:既追求稳定性，又不过分追求,且DevOps成熟度基本满足业务需求)

...
展开

作者回复: 某条业务线是否推行SRE，有个判断依据就是，是不是需要运维或SRE这样的团队介入，如果需要运维或SRE保障，那就要推行后面我们要讲的SRE体系，如果是开发自己维护，没有另外的团队参与，那就自行判断。

关于你的理解，我觉得很棒，特别是B，任何一个岗位或方法的出现，都是因为有问题解决不了被倒逼出来的，从我的角度，DevOps是如此，SRE也是如此。

潜行
2020-03-21

我们是只有三个人的运维小团队，而且职能也没有分的那么清楚细，基本上是每个人所有的事情都做，而SRE又涉及到这么多方面。请问老师，对于我们这种小团队，要实践SRE应该先从哪方面入手?

作者回复: 团队规模不大，我不建议一开始就去推行SRE。按照经验，这个阶段很有可能你的运维标准，自动化这些工作还没有完全做到位，如果要入手，建议从这里开始，再往后就是做软件的发布，再接下来可以考虑按照我们后面要讲的SLO的内容，去做稳定性的运营。

Helios
2020-03-20

虽然现在是在做devops，我们公司没有sre的岗位，听完这节课之后还是对这两个岗位有了一定的认识。
devops目标是提升开发效率和提升交付效率。
sre是保证服务稳定。

devops针对的是交付产品和开发者，sre针对的是服务。...

展开

作者回复: 不仅toC需要，toB也需要，很多的云计算公司，都有专门的SRE部门和岗位。

再就是也不一定非要有SRE的岗位，就像我在本篇中讲到的，很多的SRE工作我们都在做，只不过是分散在了不同的部门和岗位上。

1

硕

2020-03-20

传统的运维很容易变成开发的工具人。sre 通过自动化系统、制定规范流程等工资将自己脱离工具人角色，再运用运维体系丰富的知识结合当前线上的实际情况，尽可能提升线上稳定性

作者回复: 除了开发工具，还可以通过线上系统运行的数据来运营，比如整体比定性指标、资源利用率、成本分布等等。

OlafOO

2020-03-20

MTTF，还有种缩写是Mean Time To Failure ，平均故障前的时间，和文章里的MTBF一个意思。可用性的计算公式MTTF/MTTR+MTTF 是稳定性领域的第一性原理，，这个在左耳听风的专栏弹力设计里也有讲到。。。缩写方式太多😂

展开

作者回复: 掌握住一种就可以，后面我会详细讲应该如何计算，环境继续学习。

张春阳

2020-03-19

devops只是sre中的一环，提供高效的交付，测试，发布。

作者回复: DevOps是从用户价值交付视角出发，SRE是从稳定性视角出发。

春来草自青
2020-03-19

无论sre还是devops，从实用的角度看，哪个能够更简单高效低成本地解决眼下的痛点，并为以后的技术演进保留足够的想象空间，就选哪个。行业不缺牛人，也不缺顶尖技术，但每家公司的条件和能负担的成本是不一样的，设合自己的才是最好的

展开

作者回复: 非常正确，适合自己的才是最好的，一定要关注自己的问题，而不是仅仅关注各种Buzz Word。

shaonbean
2020-03-18

更看好SRE

展开

leslie
2020-03-18

DevOps的课程其实自己学习的蛮深包括相关书籍可以说不少都看了一遍，相关工具如果要花费大量时间可能对我而言确实有点难，尤其运维【注：我所值的运维不仅仅是指系统，包括数据库、操作系统、机房。。。这个整体】
不过开发能力的短板对我而言确实短时间难以提升太多：效率和平台体系的合理结合，如何二者互补我觉得才是更加去追求的。...

展开

作者回复: “过分去追求SRE或者DevOps，可能我们反而会迷失其本意。”，这句你抓住重点了，最重要的是看问题是什么，而不是纠结SRE或DevOps是什么，而是看他们能怎么帮助我们解决问题。

还有最后，非常期待和希望，你能在课程结束时，分享出你自己的感受和内容